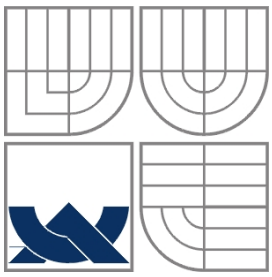
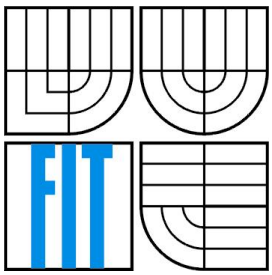


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STEGANOGRAFIE
STEGANOGRAPHY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Jiří Podhorský

VEDOUCÍ PRÁCE
SUPERVISOR
Doc. Dr. Ing. Pavel Zemčík

BRNO 2011

Abstrakt

Tento bakalářský projekt se zaměřuje na steganografii a prezentuje nové modifikace známých metod steganografie a jejich implementace. Též prezentuje implementaci adaptivní metody pro srovnání, metody "sousedů". Obě tyto metody jsou založené na využití nejméně významného bitu. Také je ukázáno, že nově navržená modifikace může být použita i v jiných metodách steganografie. Smyslem navržené metody je zakódovat tajnou informaci do digitálního obrázku, ve kterém nebude informace lehce detekovatelná. Metoda používá pseudonáhodný generátor pro generování pozic modifikovaných pixelů. Tajná informace je převedena z 8 na 10 bitů, aby byla vyrovnána pravděpodobnost nalezení 0 a 1 zakódované do obrazu, pro snížení zjistitelnosti. Zakódovaná informace může být kódována jednoduchou kryptografickou metodou, pro zvýšení bezpečnosti vložené informace. Tento bakalářský projekt vysvětluje tuto metodu, její implementaci a prezentuje experimenty a výsledky.

Abstract

This bachelor project focuses on steganography and presents new modifications of known metods steganography and their implementation. It also presents implementation of adaptive metod Neighbours, for comparison. Both of the metods are based on the „Least Significant Bit“ exploitation. It is also demonstrated that newly proposed modification can be used in other metods of steganography. The purpose of proposed method is to encode a secret information into the digital image so that the information is not easily detectable. The metod uses a pseudo-random generator for generation of positions of pixels to be modified. The secret information is converted from 8 bit to 10 bit in order to equalize the probability of occurrence 0 and 1 encoded into the image to reduce detectability. The information can be encoded with a simple cryptography method to increase security of embeded information. The bachelor project explains this method, its implemenation and it presents some experiments and results.

Klíčová slova

Steganografie, Vodotisk, Digitální obrázky, Metoda LSB, Metoda sousedů

Keywords

Steganography, Watermark, Digital images, LSB metod, Neighbors Method

Citace

Jiří Podhorský: Steganografie, bakalářská práce, Brno, FIT VUT v Brně, 2010/2011

Název bakalářské práce v jazyce práce

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením...
Další informace mi poskytli...
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jiří Podhorský
21.1.2011

Poděkování

Děkuji touto cestou vedoucímu práce, který mi velmi pomohl s formátováním a seskupením textu a s vymyšlením vlastní prezentované steganografické metody.

Obsah

1. Úvod

Tato bakalářská práce pojednává o steganografii, o metodě pro přenášení tajných dat, která nešifruje data pomocí šifry jako je tomu u kryptografie, ale která tajná data skrývá. Tato metoda byla vynalezena v dávném starověku a od té doby se neustále zdokonalovala, až do dnešní doby. O tomto postupném zdokonalování steganografie a ukazování některých jejích metod v různých časových dobách je tato práce. Dále tato práce totiž obsahuje přehled nejznámějších používaných metod dnešní steganografie, které informace ukrývají v digitálních médiích. Hlavní část této práce ale spočívá ve vytvoření vlastní steganografické metody, jejím návrhu, implementaci a srovnání s metodou sousedů.

Digitální média mohou být např. obrázky, multimédia, hudba, zvuk, které jsou v číslicové(digitální) podobě uloženy na pevném záznamovém zařízení. Pro jejich spuštění je potřeba zařízení, které je schopno s těmito digitálními daty pracovat a zobrazit je ve vizuální podobě. Nejčastěji využitím počítačů, které reprezentují univerzální zařízení, ale je možné je přehrávat i za pomoci specializovaného hardwaru, například mobil, mp3 přehrávač atd... V těchto digitálních médiích jsou ukryty informace tak, že jejich ukrytí není možné rozpoznat lidskými smysly, především zrakem a sluchem.

V této práci se dozvíte více o této metodě skrývání informací, o vzniku steganografie, její vývoj od dob starověku do nynějších dob. Také zde bude probírána implementace konkrétní aplikace, která ukazuje moderní metody steganografie. Tato aplikace bude poté bude testována na spoustě obrázků různých formátů a budou pečlivě analyzovány změny mezi originálním obrázkem a obrázkem, který skrývá tajnou informaci.

O metodách LSB a transformaci domény, o vzniku steganografie a o alternativní technologii – vodotisku se můžete dočíst ve 2 kapitole. S touto kapitolou souvisí kapitola 3, ve které shrnuji informace z kapitoly 2 a zamýšlím se nad nimi. Ve 3 kapitole také tvořím požadavky pro vytvoření vlastního návrhu steganografické metody. Tento návrh metody a implementace aplikace, používající tuto metodu, je obsaženy ve čtvrté kapitole. Následuje kapitola 5, ve které experimentuji s vlastní aplikací a ukazuji výsledky těchto experimentů. Poté zbývá pouze závěr ve kterém shrnuji celou svou práci. Obrázky, které nebylo možné vložit do kapitoly 5, se nachází v příloze A.

2. Historie steganografie a její nynější stav

V této kapitole se probírá obecný význam steganografie, její praktické použití a princip. Dále přehled vývoje steganografie od dob před Kristem až do nynější doby pomocí metod a principů, které byly používány. Poté do doby elektroniky, ve které se ocitáme nyní a ve které je nutno používat jiné metody. Tyto metody jsou v této kapitole taktéž zachyceny. Toto není encyklopedie, nejsou zde uvedeny všechny metody a informace, ale pouze ty, které jsou zajímavé a ze kterých je možné si udělat představu o principu a fungování steganografie. Praktickou představu steganografie poskytne část nazvaná *Problém vězňů***.** Nejznámější konkrétní metody steganografie jsou *LSB metody* a alternativní LSB metody, *metody sousedů a PVD metody* a nakonec *transformace domény.* Dále zhodnocení různých vlastností těchto metod a nakonec *vodotisk*, který je velmi podobný steganografii a který je možné potkat na každém kroku, aniž víme, že se jedná o vodotisk. Pro znalé steganografie bude jistě zajímavá podkapitola o *detekci steganografie***.**

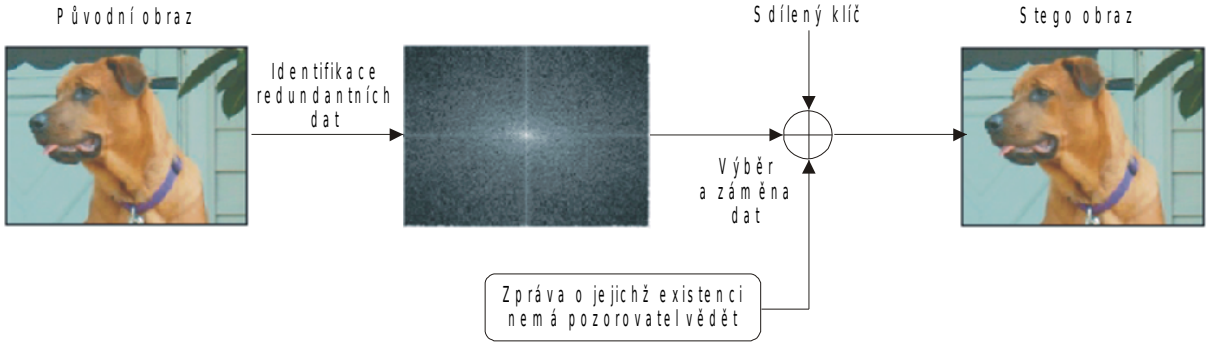
Název steganografie pochází z řečtiny [1], kde slovo *stegos* znamená *střechu* a *graphos* znamená *zobrazování*. Zahrnuje velkou skupinu metod pro skrytou komunikací.

Některé jiné metody komunikace používají šifrování (kryptografii). U těchto metod je možné zprávu zachytit, ale nikoli dekódovat. Naproti tomu steganografie skrývá účastníky této komunikace i tuto komunikaci samotnou. Respektive, tato komunikace probíhá všem na očích, ale přesto jí není možné detekovat. A jak pak lze dekódovat něco, o čemž nevíme, že to probíhá?

Pokud by ale byla komunikace přesto nějakým způsobem odhalena, bylo by ji možné přerušit nebo přečíst. Proto se steganografie někdy kombinuje s kryptografií.

2.1. Použití steganografie

Obrázek 2.1 názorně zobrazuje ukrytí tajných informací do původního obrazu použitím steganografie a kryptografie.



Obrázek 2.1: ukrytí tajných informací do původního obrazu

Skrytou informaci lze do běžných věcí, jako jsou obrázky, video nebo zvukový soubor, ukrýt pomocí steganografie tak, že nelze poznat, že v je daném souboru (médiu) něco jiného, než by mělo být. Pokud by se tyto média hloubkově analyzovala hloubkovými analýzami [4], o kterých budu psát dále a zároveň by byla k dispozici původní verze média, pak je možné detekovat změny a tím i zjistit šum v médiu. Ovšem nemohlo by být zjištěno, zda jsou to tajná data nebo zda je to jen poškození souboru v důsledku různých úprav a manipulací.

Aby nebylo možné rozlišit šum a ukrytá data, existují různé metody na jejich skrytí. Moderní steganografii je možné detekovat jen tehdy, pokud je známa skrytá informace nebo tajný klíč (postup), kterým byla data ukryta.

Nejoblíbenější skupina metod steganografie jsou patrně metody pro ukrývání zpráv do obrázku, protože obrázky jsou nejvíce rozšířené na internetu. Obrázek, stejně jako zvuková média a multimédia, snese jistou degradaci dat (pixelů), aniž by to bylo na obrázku patrné a lidské oko tuto degradaci respektive šum nemůže odhalit, protože není uzpůsobeno pro drobné detaily.
V obrázku mohou být data **komprimována** nebo **nekomprimována**.
Obrázky s komprimačními metodami [1] jsou v dnešní době většinou ve formátu JPEG. Ostatní obrázky používají ukládání dat v surové podobě. Bývá použito 8 bitů pro každou barvu, někdy i více. Tedy 24 bitů pro barevný obraz a 8 bitů pro šedě tónovaný obraz.

Kromě steganografie a kryptografie ještě existuje metoda známá jako vodotisk (watermark).
O něm se zmiňuji na konci této kapitoly.

Komunikaci pomocí steganografie lze nejlépe demonstrovat na následující pasáži, na tzv. Problému vězňů.

Problém vězňů

Dva vězni se spolu pokoušejí dorozumět [4], jak by se mohli dostat z vězení. Bohužel jediná cesta, jak se dorozumět je přes jejich dozorce. Dozorce nedovolí běžnou ani šifrovanou, respektive jakoukoli viditelnou komunikaci obou vězňů. V případě jakéhokoliv podezření je může izolovat od sebe úplně, a pak už by nebylo možné se domluvit. Vězni musí tedy komunikovat skrytě, tedy třeba právě s použitím steganografie. Příkladem této komunikace může být, že jeden z vězňů nakreslí nevinně vypadající obrázek s použitím různých barev a pošle jej druhému vězni přes dozorce. Dozorce si obrázek zkoumavě prohlédne, ale zdá se mu, že obrázek je jen umění a že se nemůže nic stát, pokud ho předá. Při předávání ovšem netuší, že například barvy na obrázku skrývají určitou zprávu. Ovšem mohou nastat problémy, které mohou zkomplikovat útěk vězňům z vězení. Dozorce může obrázek úmyslně nebo neúmyslně poškodit. Také může vytvořit falešnou zprávu a doručit ji jednomu z obou vězňů, vydávajíc jí za zprávu od druhého. Tyto problémy se také mohou ve steganografii vyskytovat.

Ve steganografii jsou věci, které je potřeba pojmenovat [4]. Je jím **originální objekt (médium)**, do kterého bude poté vložena **tajná zpráva** použitím **steganografického klíče** a tím se vytvoří **steganografický objekt (médium)**.

Praktické využití steganografie

- Mrtvá schránka (dead drop) – termín [6] používaný pro popis anonymní komunikace mezi dvěma entitami, která je využívána mezi agenty. Máme-li dva agenty, kteří se neznají, ale potřebují spolu anonymně komunikovat, lze využít mrtvou schránku. Tento termín se používá také pro to, když na určitém místě (třeba v parku), zanechá některý člověk nějakou věc (tašku, brašnu) a druhý, který ví, že tam bude ukryta, si ji vyzvedne. Ve steganografii to funguje podobně. Lze ukládat informace do určitých, nevinně vypadajících obrázků na různých serverech internetu, kde je podobných obrázků plno a nikdo nebude tušit, že některý obsahuje tajnou zprávu.

- Další a hojně užívanou možností je ukrýt informaci do obrázku, tzv. Avataru, který má každý uživatel u svého účtu. Takto je komunikace mezi odesílatelem a příjemcem naprosto anonymní.

2.2. Vývoj steganografie od historie po současnost

První zmínky o steganografii jsou v knihách od řeckého historika Herodota [4]. Xerxes poslal posla s voskovou tabulkou s nějakou zprávou, kde skutečná tajná zpráva byla vyryta do dřeva pod voskem.

Další je o řekovi Histaiaeunovi, který poslal zprávu tak, že oholil otrokovi hlavu, zprávu mu na ní nechal vytetovat, až otrokovi vlasy dorostly, poslal jej k příjemci, kde mu byly opět vlasy oholeny a tajná zpráva přečtena. Tento způsob byl ale velice zdlouhavý.

Ovšem objevitel moderní kryptografie [4] byl mnich Johannes Trithemius, který napsal trojdílný cyklus knih, „Steganographia“, kolem roku 1500, kde popsal rozsáhlý systém pro zasílání tajných zpráv, které byly ukryty v nevinně vypadajícím textu.
Například jedno z jeho schémat bylo ukrytí zprávy v dlouhém textu, obsahující jména andělů.
Tajná zpráva byla ukryta ve slovech tak, že se muselo číst v každém sudém slově, každé sudé písmeno, jak je vidět na *příkladu1.* Tím dostali tajnou zprávu.

padiel a**p**o**r**s**y** mesarpon o**m**e**u**as peludyn m**a**l**p**r**e**a**x**o
Příklad 2.1: Steganografie - informace ukryté ve jménech andělů

Po přečtení těchto jmen popsaným způsobem vznikne zpráva: „prymus apex“

Další metody:

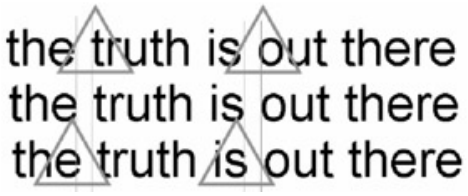
- **Mikrotečka** - princip spočívá v tom, že se nejprve obrázek zprávy zmenší na velikost poštovní známky. Poté se reverzním mikroskopem zmenší tento obrázek na pouhý bod. Dále se vyvolá negativ a obrázek se vytiskne na film. Bohužel tato metoda potřebuje speciální vybavení.

- **Nulové šifry (Null Ciphers)** – metoda [4] spočívá v tom, že nezašifrovaná zpráva se ukryje do větší, nevinně znějící zprávy. Je to poměrně známá metoda, kde pro dešifrování je nutné číst n-té písmeno každého slova zprávy. Tyto písmena dávají dohromady hledanou tajnou zprávu.

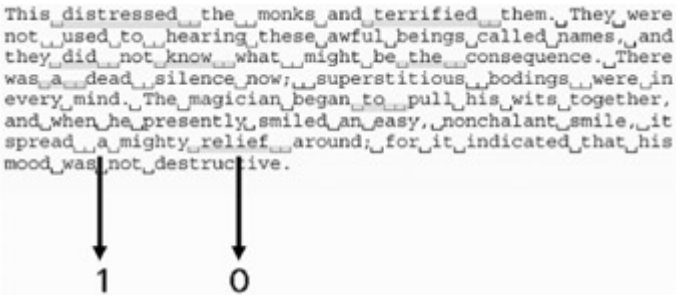
- **Acrostics** - metoda [4] je podobná předešlé metodě nulové šifry. Tajná zpráva je ukrytá do básně nebo do sekvence řádků. Tajná zpráva je obsažena obyčejně v písmenech na začátku každého řádku.

- **Novinový kód (newspaper code)** - tento kód [4] se používal hlavně v době rozmachu novin, kdy noviny měl prakticky každý. V novinách, nad určitými písmeny byly vytvořeny dírky. Tyto písmena, čtená po sobě, dávají tajnou zprávu. Později byl použit místo dírek neviditelný inkoust, který zlepšil účinnost této metody a tím zhoršil schopnost čtení tohoto kódu nepovolané osobě.

- **Systém psaní mezer a odsazení (Type Spacing and offsetting)** - k zakódování zprávy se používá mezer mezi slovy [4], kde některá písmena lehce vybočují ze svých normálních pozic. Tyto písmena dávají dohromady tajnou zprávu. Je možné použít místo jednotlivých slov celé bloky slov. Toto je ukázáno na *obrázku 2.2.*
  Druhý způsob je použití redundantních mezer mezi slovy a na základě jejich množství mezi jednotlivými slovy dekódovat informaci v binární (dvojkové) formě. Následující *obrázek 2.3* ukazuje použití mezer



Obrázek 2.2: Použití odsazení písmen



Obrázek 2.3: Použití redundantních mezer

- **Neviditelný inkoust(Invisible ink)** - je to vlastně bezbarvá tekutina [4], která působením tepla, světla nebo speciálních chemikálií změní barvu a učiní text viditelný. Účinky podobné neviditelnému inkoustu mají mléko, ocet, citrónový džus a moč, ale k výrobě pravého neviditelného inkoustu je potřeba složitých procedur. Příklady receptů neviditelného inkoustu:

  Kobalt oxidu rozpuštěný v kyselině chlorovodíkové nebo v kyselině dusičné produkuje kapalinu, která je neviditelná, dokud není zvednuta k plameni, který způsobí její modré svícení. Modrá pak měla zmizet, pokud na ní foukáme.

  I vejce byla použita pro skrytí zpráv. Zpráva je napsána na skořápku čistého vejce, a pak inkoust projde porézním povrchem skořápky. Když je vejce uvařeno na tvrdo, skořápka se opatrně oloupe a odhalí se zpráva.[4]

- **Grilles (Cardano's Grille)** - princip této metody [4] je jednoduchý. Každý příjemce má kousek papíru nebo lepenku obsahující otvory. Tato děrovaná lepenka je přiložena na nevinně vypadající zprávu a díry ukazují písmena, které čteny po sobě dávají dohromady tajnou zprávu. Protože nevinně vypadající zpráva může být třeba i kniha, nepovolaný, který by dostal do ruky tuto kartičku s otvory, nemůže vědět, na kterou stránku které knihy ji přiložit, aby získal onu tajnou zprávu.

2.3. Nové metody steganografie

V našem elektronickém světě jsou prakticky tři způsoby[4] ukrytí ukrýt digitální zprávy.
Buď *vstřikování (injection), substituce (substitution)* a *generování nových souborů.*

- Metoda *vstřikování* - vkládání skryté zprávy přímo do určitého média (souboru). Problémem je, že konečný soubor je obvykle větší než originální soubor a to jej dělá lépe odhalitelným.
- Metoda *substituce* – data na médiu (souboru) jsou nahrazena tajnými daty. Jelikož dochází k nahrazení, nedochází k vysokému nárůstu velikosti souboru. Bohužel v závislosti na množství skrytých dat dochází k degradaci kvality výsledného souboru.
- Generování nových souborů – dříve to byla metoda použitá v „problému vězňů“. Nyní je to metoda vkládání tajné zprávy do vygenerovaného spamu, který je pak odeslán příjemci.

O těchto metodách je pojednáváno v následujících podkapitolách.

2.4. LSB (Least Significant Bit)

Je to technika substitučního systému – dochází k vkládání do souboru. Tuto metodu [4] nelze použít v komprimovaném obrázku. Jedná se o nahrazení nejmenšího bitů (LSB) každého pixelu obrázku jiným bitem. Bitem, který tvoří tajnou zprávu. Nemusí se jednat pouze o obrázek. Je možné použít i video nebo audio, u kterých je to podobný způsob. Tento způsob vytváří jisté odchylky v obraze, respektive jistý šum, který je nevýznamný a nepostřehnutelný pouhým okem. Princip této metody je ukázán na *příkladu2*.

Máme 4 byty do kterých potřebujeme zakódovat tajnou informaci - 1010

Tyto byty před vložením:	Tyto byty po vložení:	Postup:
10001001	10001001	Tady je 1 obsažena, neděláme nic
11011010	11011010	Tady je 0 obsažena, neděláme nic
10101010	10101011	Tady není 1 obsažena, změníme 0 na 1
00001011	00001010	Tady není 0 obsažena, změníme 1 na 0

Příklad2: Ukázka metody LSB

Musíme měnit jen LSB bit, vyšší byty by produkovaly vyšší odlišnost bytu od původního a tím např. na obrázku úplně jinou barvu.
Pokud písmena kódujeme v ASCII abecedě, jedno písmeno lze uložit do 8 bitů. Každý bit lze uložit do 1 pixelu u šedě tónovaného obrázku a až 3 bity do 1 pixelu u barevného RGB obrázku.
Například do obrázku o rozměrech 720 a 480 lze uložit až 43200 znaků v šedě tónovaném obrázku a 129600 znaků v barevném obrázku.
Existuje i vylepšení této metody, kde se ukrývá informace také do barevných složek obrazu, ale ne v barevném modelu, v kterém je ukládán, ale v transformovaném barevném modelu například YUV. Po uložení informace se model přetransformuje zpět do původního barevného modelu. Takto vznikne obrázek s lépe ukrytou informací.

2.5. Metody sousedů

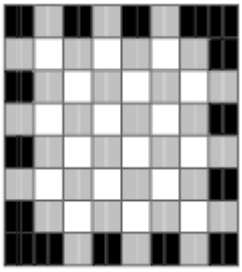
Základní nevýhodou metody LSB je, že všechny pixely obrázku nemohou vydržet stejné množství změn, respektive u některých pixelů je změna patrnější než u jiných, a pak může docházet k narušení obrázku, a tím ulehčit detekci skrytého obrázku nepovolaným osobám.

K získání lepší kvality obrázku se zakódovanou informací metodou LSB ***Chan K.*** a ***Cheng M.*** navrhli proces pro optimalizaci pixelů jejich přizpůsobením (Optimal Pixel Adjustment Process – OPAP). Tato metoda je aplikována na steganografický obraz tak, aby byl výsledný pixel blíže k původní hodnotě.

Metody OPAP a LSB ale nejsou adaptivní, proto ***Wu C.*** a ***Tsai H.*** navrhli adaptivní metodu[3] založené na vztazích mezi pixely, kde je počet vložených bitů do každého pixelu proměnný. Toto výrazně zvýšilo kvalitu obrázku. Následující tři metody využívají okolí jednotlivých pixelů pro odhadnutí množství dat tajné informace, které mohou být vloženy do jednotlivých pixelů, bez toho, že by se obrázek deformoval. Tyto navržené metody využívají tzv. psycho-vizuální redundance a závislosti na pixelu okolních sousedů. Korelace mezi pixelem a jeho sousedy rozhodne, zda bude umístěn v „hladké oblasti“ nebo „komplikované oblasti“. Hladká oblast je místo na obrázku, kde mají pixely se svými sousedy stejné, nebo podobné barevné hloubky. Komplikovaná oblast je oblast, kde mají mají pixely se svými sousedy velmi odlišné barevné hloubky.

- ***Metoda čtyř sousedů***

Tato metoda [3] kontroluje korelaci mezi pixelem a horním, spodním, pravým a levým pixelem. Vlastně funkce této metody připomíná šachy. Každý druhý pixel zleva doprava od druhého řádku až do předposledního je kontrolován. Funkce metody je ukázána na *obrázku 2.4*.



Obrázek 2.4: Vizuální zobrazení funkce metody

$$(1)\quad d_i = (p_{nahoře} + p_{dole} + p_{vlevo} + p_{vpravo})/4 - p_i$$

$$(2)\quad n = floor(\log_2 |d_i|)$$

$$(3)\quad d_i' = 2^n + b \quad pokud\ je\ \ d_i\ \ kladné,\ \ d_i' = -(2^n + b)\ \ jinak$$

Bílé pixely reprezentují pixely, do kterých jsou data vložena, **šedé** ty, které zůstávají nezměněny. Jsou to sousedské pixely, které rozhodnou o množství vložené informace. Zbývající **černé** pixely jsou neobsazené a nepřispívají do procesu vkládání. Rozdíly hodnot d_i (1), indikují hladkost nebo komplikovanost (drsnost) oblasti.

Malé hodnoty d_i indikují, že region 3x3 je hladký. Jsou to hodnoty od 0 do 7. Do pixelů, v hladké oblasti lze uložit pouze 3 bity. Větší hodnoty d_i naznačují, že pixel spadá do drsné oblasti. Proměnný poměr určuje, kolik bitů tajných dat je vloženo (2), minimálně 3.
n bitů dat je načteno z tajných dat, které chceme vložit, a převedeno na svůj decimální ekvivalent, kterému se říká ***b*** . Přidáním k nejnižší mezi 2^n, se vytvoří nový rozdíl hodnot d_i', který leží na stejném rozsahu $[2^n, 2^{n+1}]$ jako d_i (3). Někdy může nová hodnota pixelu p_i vypadnout z rozsahu [0, 255], tedy v 8 bitech, ve kterých je zakódován 1 pixel. Takový pixel nesplňuje podmínky (4),(5), proto se do něj informace nevkládají.

$$(4)\quad když\ d_i \geq 8\ a\ zároveň\ \frac{p_{nahoře} + p_{dole} + p_{vlevo} + p_{vpravo}}{4} < 2^{n+1} - 1,$$
$$pak\ pi' = pi\ ,\ respektive\ \ pixelu\ předávám\ novou\ hodnotu$$

$$(5)\quad když\ d_i < 8\ a\ zároveň\ \frac{p_{nahoře} + p_{dole} + p_{vlevo} + p_{vpravo}}{4} + 2^{n+1} > 256,$$
$$pak\ steganografický\ pixel\ odpovídá\ originálnímu\ pixelu$$

$$(6)\quad p_i' = \frac{p_{nahoře} + p_{dole} + p_{vlevo} + p_{vpravo}}{4} - d_i'$$

Dále se vypočítá odpovídající pixel steganografického obrázku (6). Následné zjištění informace z

pixelu je obrácený postup k předchozímu vkládání informace. Rozdíl hodnot d_i^* je vypočítán pro každý bílý pixel podobně jako u (1). Skrytá informace *b* je získána z d_i^* (7).

$$(7)\quad b = d_i^* - 2^n \quad pokud\ je\ \ d_i^*\ \ kladné\,,\ \ b = -\left(d_i^* + 2^n\right)\ \ jinak$$

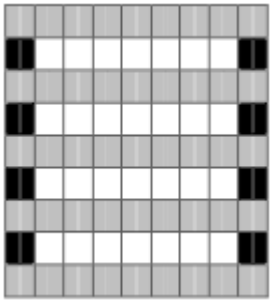
b je pak převedeno do jeho binární reprezentace, která se skládá z *n* bitů. Předešlé podmínky (1),(2) jsou použity pro určení, který pixel obsahuje skrytou informaci.

Další metody

metoda diagonálních sousedů a **metoda osmi sousedů** [3] se od **metody čtyř sousedů** [3] příliš neliší. Jsou použity pouze jiné výpočty rozdílu mezi pixely a jiné vyhodnocování sousedů pixelů. Opět bílá políčka znázorňují pixely, které se budou měnit. Šedá ty políčka, u kterých se bude určovat, zda jsou hladké nebo drsné. A černá ty pixely, které nebudou použity.

- ***Metoda diagonálních sousedů***

  Tato metoda [3] se vyznačuje tím, že hladká i drsná oblast v originálním obrázku je založena na vztazích pixelů s jeho diagonálními sousedy (8), jak je to ukázáno na *obrázku 2.5*. Ostatní rovnice jsou stejné jako u metody čtyř sousedů.

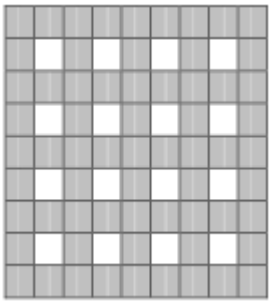


Obrázek 2.5: Vizuální zobrazení funkce metody

$$(8)\quad d_i = \frac{p_{nahoře\,vlevo} + p_{nahoře\,vpravo} + p_{dole\,vlevo} + p_{dole\,vpravo}}{4} - p_i$$

- ***Metoda osmi sousedů***

  Tato metoda [3] používá všech 8 sousedů v oblasti 3x3 (9), jak je to ukázáno na *obrázku 2.6*. Ostatní rovnice jsou stejné jako u metody čtyř sousedů. Vyznačuje se mnohem větší přesností, než obě předešlé metody.



Obrázek 2.6: Vizuální zobrazení funkce metody

$$(9)\quad d_i = \frac{p_{nahoře} + p_{dole} + p_{vlevo} + p_{vpravo} + p_{nahoře\,vlevo} + p_{nahoře\,vpravo} + p_{dole\,vlevo} + p_{dole\,vpravo}}{8} - p_i$$

Tato rovnice je vlastně spojením obou rovnic předešlých metod.

2.6. Metoda PVD (Pixel-value differencing)

V této metodě [2] šedě škálovaný originální obrázek je rozdělen do nepřekrývajících se bloků, tvořených dvěma po sobě jdoucími pixely p_i a p_{i+1}. Rozdílová hodnota d_i se z každého bloku vypočítá odečtením p_i od p_{i+1}. Množina všech rozdílových hodnot by měla být v intervalu <-255,255>. Hladká oblast má malé hodnoty | d_i |, ostrá oblast má vyšší hodnoty | d_i |. Lidské oči zvládnou tolerovat více změn v ostré oblasti než v hladké. Proto prvním krokem metody PVD je vytvořit tabulku rozsahů s n souvislými řadami R_k, kde k = 1,2,3,...,n. A rozsah tabulky je od 0 do 255. Jiný rozsah je v rozsahu R_k Î [l_k, u_k], kde l_k je menší rozsah než 0 a u_k je větší rozsah než 255. Šířka w_k se spočítá $w_k = u_k - l_k + 1$. Tato šířka určuje, kolik bitů může být uloženo v dvou, po sobě jdoucích pixelech. K získání tajných dat je potřeba originální tabulku rozsahů.

Originální verze PVD metody[2] předpokládá skryté data v dlouhé řadě bitů a že originální obrázek je šedě škálovaný. Toto je vkládací algoritmus:

1) Vypočítá se rozdílová hodnota d_i mezi dvěma, po sobě jdoucími pixely p_i a p_{i+1} pro každý blok originálního obrázku - $d_i = p_{i+1} - p_i$.
2) Pro vyhledání vhodného R_k v tabulce rozsahů použijeme | d_i | tak, že spočítáme nejdříve $j = min(u_k - |d_i|)$, kde $u_k \geq [d_i] pro\ všechny\ 1 \leq k \leq n$. Pak R_j je hledaný rozsah.
3) Vypočítáme množství bitů skrytých dat t, které je možné vložit do každého páru dvou po sobě jdoucích pixelů. t se odhadne z šířky w_j, $t = [\log_2(w_j)]$.
4) Načteme t bitů z binárních dat a převedeme bitovou sekvenci na desítkovou hodnotu b. Například, je-li bitová sekvence 101, pak je převedená hodnota b=4.
5) Vypočítáme novou hodnotu d'_i. Je-li $d_i >= 0$ $d'_i = l_j + b$, je-li $d_i < 0$ $d'_i = -(l_j + b)$. Touto hodnotou přepíšeme dřívější rozdíl d_i.
6) Modifikujeme hodnoty p_i a p_{i+1} formulí: $(p'_i, p'_{i+1}) = (p_i - [m/2], p_{i+1} + [m/2])$, kde $m = d'_i - d_i$. Nakonec změnou hodnot páru pixelů (p_i, p_{i+1}) pixely(p_i', p_{i+1}') jsou data uložena.

Opakujeme kroky 1-6 dokud nebudou všechny pixely vloženy do obrázku, tím vytvoříme steganografický obrázek.

Během získávání dat z obrázku je nutná tabulka rozsahů. Stejnou metodou, jakou jsme

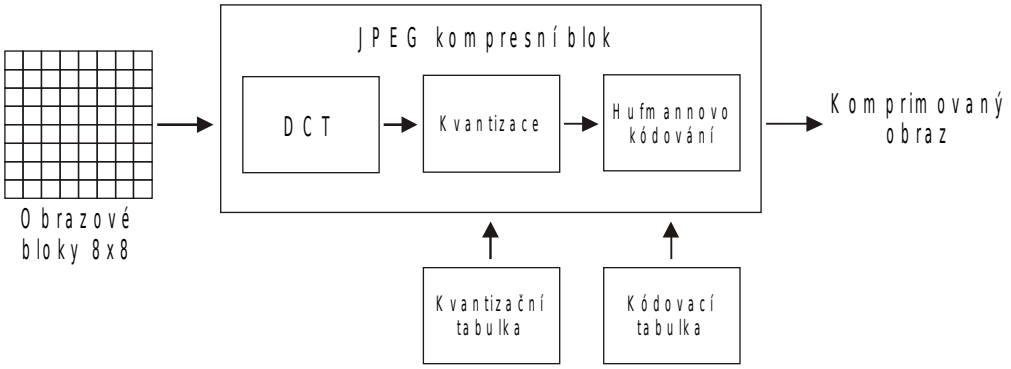
vkládali data do párů pixelů (bloků) steganografického obrázku, data opět získáme. Spočítáme rozdílovou hodnotu d^*_i pro každou dvojici po sobě jdoucích pixelů p^*_i , p^*_{i+1} steganografického obrázku. Poté pro nalezení vhodné R_j je použita $|d^*_i|$, stejně jako v předchozím kroku 2. Pak je odečtením l_j od $|d^*_i|$ získána hodnota b^*. Pokud nebyl steganografický obrázek změněn, b^* je roven b. Nakonec je dekadická hodnota převedena na binární sekvenci t bitů, kde $t = [\log_2(w_j)]$.

2.7. Transformace domény

Transformace domény je běžně používaná metoda [1] ukládání dat, například ve formátu JPEG. Obecně řečeno, JPEG (Joint Photographic Experts Group) je komprimační metoda, která se zbavuje nadbytečných informací ve frekvencích, které jsou lidskému oku neviditelné. JPEG soubory jsou nejpoužívanější formáty na internetu, protože jeho obrázky mají malou velikost.

Dříve se myslelo, že do JPEG nebude možné ukládat informace, protože ztrátová komprese by vedla ke změně skryté informace a ztrátě, při každé kompresi i dekompresi. Bylo nutné vymyslet metodu, aby skrytá informace byla stále dostupná. Tato metoda spočívá v tom, že během DCT se v koeficientech objevují zaokrouhlovací chyby. Proto se tyto koeficienty dají využít jako pixely, do kterých se uloží tajná informace jako do nejnižších bitů pixelů u LSB algoritmu. Tady se ovšem použije LSB algoritmus také, využijí se nejnižší byty těchto koeficientů a následně jsou koeficienty pak zakódovány bezeztrátovým algoritmem. Toto je ještě bezpečnější, protože koeficienty nejsou ve viditelné části obrázku a proto je je velmi obtížné detekovat

Při kompresi obrázků do JPEG formátu musíme nejdříve převést jejich RGB model do YUV modelu. V této reprezentaci Y vyjadřuje svítivost (nebo jas), U a V vyjadřují barevný rozdíl (nebo barvu). Je známo, že lidské oko je více citlivé na změny svítivost (nebo jas) pixelu než změny barvy pixelu. Tohoto je využíváno u JPEG komprese - pomocí snížení vzorkování dat snížíme velikost výsledného obrázku. Komponenty barvy, tedy U a V jsou v horizontálním a vertikálním směru rozpůleny, proto bude výsledný soubor poloviční. Poté se pro JPEG použije *Diskrétní cosinová transformace (DCT)*. Může se také použít podobná *Diskrétní Furierova Transformace (DFT).* DFT transformace transformuje signál z obrazu na frekvenční reprezentaci, pomocí seskupení pixelů do bloků 8x8 pixelů, a poté jejich transformace na 64 DCT koeficienty. Modifikace jednotlivých DCT koeficientů ovlivňuje 64 pixelů v bloku. Lidské oko vidí dobře malé rozdíly v jasu ve větších oblastech. Ale nedokáže rozlišit rozdíly ve vyšších frekvencích jasu. Proto lze snížit vysoké frekvence bez změny vzhledu obrázku. JPEG toto provede snížením všech hodnot v bloku kvantizačním koeficientem na polovinu. Výsledné hodnoty jsou zaokrouhleny na celá čísla a koeficienty jsou zakódovány Huffmanovým kódováním, které dále sníží velikost obrázku.



Obrázek 7: Vizuální zobrazení funkce metody

Další metoda využívající transformaci domén je **Patchwork** (Slátanina)
Patchwork [5] používá kódování pomocí redundantních vzorů pro vložení informace do obrázku. Algoritmus přidává redundanci do skryté informace a tu potom rozptýlí po obrázku. Použije se pseudonáhodný generátor pro vybrání dvou oblastí v obrázku, místo α a místo β. Všechny pixely v místě α jsou zesvětleny, zatímco v místě β ztmaveny. Nevýhodou patchworku je vložení pouze jednoho bitu. Další bity mohou být vloženy při rozdělení obrázku na pod-obrázky a vložením bitu do každého podobrázku. Výhodou je, že tajná zpráva je rozprostřena po celém obrázku. Pokud bude jedna oblast poškozena nebo zničena, ostatní oblasti přežijí. Velikost vkládané zprávy udává, kolikrát bude zpráva v obrázku opakována. Pokud je příliš velká, bude vložena pouze jednou. Patchwork je ceněn pro svou nezávislost na cílovém obrázku. Tajná zpráva kódovaná v patchworku může dokonce přežít i převod mezi ztrátovou a bezztrátovou kompresí.

2.8 Ostatní digitální metody steganografie

- **Spread spectrum** - Tato metoda [5] je vhodná pro steganografii i pro vodotisk. Jedná se o vkládání nízkofrekvenčních signálů mezi ostatní frekvence média. Tajná informace se vloží do šumu, který byl vygenerován generátorem šumu. Pak vloží takto upravený šum zpátky do použitého média. Tato metoda je velmi odolná, a informaci je obtížné detekovat.. Také má vysokou kapacitu, pro uložení dat. Bez originálního média ji nelze detekovat.
- **Maskování vnímavosti (perceptual masking)** - Tato technika [4] využívá toho, že jeden signál (zvuk) je neznatelný, kvůli přítomnosti jiného zvuku. Běžný příklad této metody je možné spatřit ve filmech, kde potřebuje jedna osoba říci něco druhé osobě, ale ví že jsou odposloucháváni. Proto pustí sprchu, spláchnou, pustí nějakou hudbu nahlas nebo něco jiného hlučného a nevinně vypadajícího. Pak mohou mluvit šeptem, aniž by byl jejich hovor odposlechnut.
- **Statistická metoda** - Tato metoda [4] spočívá ve vkládání jednoho bitu informace do originálního obrázku. Tím se vytváří malá statistická změna. Změna indikuje 1, nezměněný obrázek indikuje 0. Ovšem příjemce musí rozlišit změněný a nezměněný obrázek.
- **Metody založené na generování obalu** - Tyto metody [4] generují obal, který ukryje skutečnou tajnou zprávu. Je to vlastně hromada vygenerovaného textu, který by měl být co nejrychleji vygenerován a nikdo neměl zájem ho číst, v kterém se ukryje zpráva pomocí různých metod. Například program Spam Mimic, který generuje spam, který výtečně skryje tajnou zprávu.

2.9. Detekce Steganografie

V této podkapitole jsou uvedeny některé typy analýz a útoků na steganografická média.

Detekce steganografie

- **Statistické testy** - tyto testy [4] odhalují, že obrázek byl modifikován tím, že statistické vlastnosti obrázku se vychýlily z normy. Některé statistické testy mohou být nezávislé na datovém formátu a měří pouze redundantní data. Obrázky se skrytými daty mají vyšší redundanci dat.
- **Slovníkové útoky** - slovníkové útoky [4] jsou řazené do metod útoků hrubou silou (brutal force attack), z nichž většina generuje postupně všechna hesla a ty zkouší. Na rozdíl od nich, slovníkové útoky negenerují všechny hesla, ale místo toho zkouší postupně hesla z nějakého listu, většinou slovníku. Slovníkové útoky se zakládají na tom, že steganografické metody

většinou skrývají hlavičku informací na začátku skryté zprávy. Tyto hlavičky většinou obsahují informace o délce zprávy, kompresní metodě a dalších důležitých věcech.

- **Viditelný šum** - útoky a analýzy [4] mohou mít několik forem: detekce, získání a zneškodnění nebo zničení skryté informace. Je důležité použít vhodné krycí médium pro tajnou zprávu a dobrou metodu pro uložení tajné informace. Některé obrázky nebo hudba mohou být vlivem uložených informací hodně degradované (zkreslené), a tím mohou prozradit ukrytí tajné informace v sobě. Pak lze použít například „útok se znalostí média“, který porovná originál obrázku s nynějším zašuměným steganografickým obrázkem a objeví odchylky. Tyto odchylky lze pak porovnat s podpisy steganografických programů a tím zjistit program, kterým to bylo zakódováno, a poté obrázek zpětně rozkódovat.

Druhy útoků

- **Pouze steganografický útok** - v tomto útoku [4] máme k dispozici jen steganografické médium (obrázek se zapsanou informací).
- **Útok se znalostí obalu** - v tomto útoku [4] máme k dispozici i steganografické médium i originální médium (obrázek) a informace zjišťujeme jejich porovnáním.
- **Útok se znalostí zprávy** – v tomto útoku [4], známe zprávu, máme steganografické médium, kde je zapsána a zjišťujeme metodu, kterou je zapsána zpráva do média.
- **Zvolený steganografický útok** – používaný [4], máme-li steganografické médium a steganografický program nebo metodu.
- **Útok se zvolenou zprávou** – steganografická analýza [4] generuje steganografické médium ze zprávy, použitím konkrétních programů. Pak se hledá podpis, který by se dal vyhledat v jiném steganografickém médiu.

Vypínací nebo také aktivní útoky

Tyto metody útoků [4] poškozují ukrytou zprávu. Jsou to vlastně úpravy obrázku.

- **Rozmazání:** vyhladí přechody a sníží kontrast zprůměrováním pixelů vedle tvrdých hran definovaných řádků a oblastí, kde jsou významné barevné přenosy.
- **Šum:** náhodný šum se vkládá náhodně barevné pixely do obrázku. Rovnoměrný šum vkládá pixely a barvu tak, že blízké se podobají originálním pixelům.
- **Redukce šumu:** redukuje šum v obrázku regulací barev a zprůměrováním hodnot pixelů.
- **Brus:** Broušení je opak rozmazání. Zvyšuje kontrast mezi přilehlými pixely, kde jsou významné kontrasty barev, obvykle na hraně objektu.
- **Rotace:** Rotace pohybuje obrázkem okolo středového bodu v dané rovině. *Převzorkování*: převzorkování zahrnuje interpolaci procesů pro minimalizování „chatrnosti“ běžně asociované rozšířením obrázku.
- **Zjemnění:** platí pro rovnoměrné rozmazání obrázku k zjemnění hran a redukci kontrastu. Způsobuje méně zkreslení než metoda rozmazání.
- **Nestabilní útok (jitter)** - Funguje stejně dobře ve steganografii i ve vodotisku. Účelem je narušit rozmístění bitů, které identifikují vodotisk (podpis vodotisku) nebo tajnou informaci, aplikováním „chvění“ do obrázku. Jak odolný je vodotisk záleží na tom, kolik chvění dokáže absorbovat. V případě křehkého vodotisku pouhé oříznutí jednoho řádku pixelů z okraje obrázku způsobí zničení vodotisku.
- **Přidaný šum** - Jednoduše přidává dodatečně nepatrný šum do obrázku pro zabránění nebo zastavení detekce vodotisku. Protože každý pixel v obrázku má jistou toleranci šumu, kde lze nahrát informaci, která zůstane stále neviditelná. Tento útok na každém pixelu nastavuje maximální hodnotu této tolerance a tím zvyšuje množství dat, se kterým se bude muset detektor pracovat, aby zjistil, kde je vodotisk uložen.

2.10. Porovnání používaných druhů steganografie

Tabulka 1 ukazuje srovnání nejpoužívanějších druhů steganografie [5].

	LSB in BMP	LSB in GIF	JPEG compression	Patchwork
Neviditelnost	Vysoká*	Střední*	Vysoká	Vysoká
Užitečná kapacita	Vysoká	Střední	Střední	Nízká
Odolnost vůči statistickým útokům	Nízká	Nízká	Střední	Vysoká
Odolnost vůči manipulaci s obrazem	Nízká	Nízká	Střední	Vysoká
Nezávislost na formátu souboru	Nízká	Nízká	Nízká	Vysoká
Nevinnost souborů	Nízká	Nízká	Vysoká	Vysoká

* - záleží na použitém zdrojovém obrázku

Tabulka1: porovnávající různé techniky, v různých formátech

Popis použitých vlastností steganografie:

- **Neviditelnost** - neviditelnost je nejdůležitějším požadavkem [5] pro steganografické algoritmy. Ve chvíli, kdy někdo zjistí, že s obrázkem bylo manipulováno, je algoritmus prozrazen.
- **Užitečná kapacita** - narozdíl od vodotisku[1], který potřebuje vkládat jen malé množství informace o autorských právech, steganografie se zaměřuje na skrytou komunikaci a s tou související kapacita, pro vkládání tajné informace.
- **Odolnost vůči statickým útokům** - spoustu steganografických algoritmů [5] zanechává podpis, když do ukrývají do médií tajná data. Ten může být lehce detekován statistickou analýzou. Proto algoritmus **nesmí** zanechávat takové označení.
- **Odolnost vůči manipulaci s obrázkem** - než obrázek dorazí ke svému cíli [5], může projít různou manipulací, jako ořezáváním, nebo rotací. V závislosti na tom, jak byla data vložena, tato manipulace může zničit skrytou informaci.
- **Nezávislost na formátu souboru** - při posílání skryté informace nemusí být dostupný určitý typ obrázku [5], proto by měl být steganografický algoritmus nezávislý na použitém médiu.
- **Nevinnost souborů** - například vyšší velikost souboru [5], než je potřeba pro uložení obrázku v určitých rozměrech v určitém formátu může vzbudit podezření.

Popis použitých metod steganografie:

- **LSB v BMP** – Pokud se vkládají [5] data do “surového” obrázku, který nebude změněn kompresí (třeba jako BMP) existuje kompromis mezi stupněm neviditelnosti zprávy a množstvím vložené informace. V BMP je možné schovat velké zprávy, ale čím více bitů informace se skrývá, tím je větší pravděpodobnost, že změny budou patrné pouhým pohledem. Používá se hlavně v případech, kdy se hledí na kapacitu přenosu a ne na utajení informací.
- **LSB v GIF** – Slabé a silné stránky[5] má podobné jako LSB v BMP. Hlavní rozdíl je ten, že soubory GIF mají jen 8 bitovou hloubku, proto množství ukryté informace je nižší než u BMP. Obrázky GIF jsou obzvláště zranitelné vůči statistickým útokům, protože zanechává spoustu podpisů v obrázku. Zároveň, pokud se vybere špatný obrázek na ukrytí informace, mohou být změny viditelné. LSB v GIF je, ale velmi efektivní algoritmus pro vkládání rozumného množství dat ve stupních šedi.
- **JPEG komprese** – Pokud se použije algoritmus [5]„transformace domény“, uložená data mají

vysoký stupeň ukrytí. Spojíme-li to s tím, jak je JPEG formát oblíbený, je to jeden z nejlepších formátů pro ukrytí informace. Je obzvláště vhodný, pokud se zasílá přes nějaké otevřené systémové prostředí, například internet.

- **Patchwork** – Největší nevýhodou je to, že může ukrývat pouze malé množství dat [5]. Naproti tomu poskytuje vysoký stupeň ukrytí dat. Jeho hlavní výhoda je ta, že je vysoce odolný proti manipulaci s obrázkem. Při oříznutí nebo otočení obrázku mohou být některá data ztracena, ale protože informace je vložena do obrázku opakovaně, většina informace přežije. Proto je nejvhodnější pro přenos malého množství citlivých dat .
- **Metoda přidaných mezer a odsazení znaků** - Tato detekční metoda [5] se odkazuje na dříve uváděnou metodu, ukrývající tajnou informaci do mezer a formátování textu. Bohužel dnešní počítače tuto dříve uváděnou metodu kódování lehce objevit. Mohou si dát zobrazit nesrovnalosti v textu, nebo dokonce znovu přeformátovat text a smazat nadbytečné mezery, čímž zničí informaci obsaženou v textu. I když v textu nebudou jen mezery mezi slovy, ale i malé mezery mezi jednotlivými písmeny, stejně moderní textový procesor na rozdíl od pouhého oka rozliší jakoukoli posloupnost mezer, i mezi jednotlivými písmeny.

2.11. Vodotisk

Vodotisk [4] se používal a používá pro označení různých věcí tak, aby bylo patrné, kdo jej vyrobil nebo komu patří. Například při klasickém tisku bankovek se označují vodotiskem, aby je nebylo možné zkopírovat. Tím vlastně ochraňují bankovky proti padělání. Dále se využívá pro prokázání autentičnosti například u dopisů nebo jiných důležitých spisů. V klasickém tisku je zalisován do papíru. Papír pak vypadá lehce větší a vodotisk se objeví, podle typu vodoznaku, buď pod klasickým světlem, nebo pod UV lampou. Metoda vodotisku byla vynalezena již v 16. století, ale rozšířila se teprve v minulém století v oborech autentifikace a ochrany autorských práv.

Nynějším digitálním vodotiskem se obvykle podepisují multimediální soubory, jako jsou filmy, zvukové nahrávky, nebo obrázky. Stejně jako u dřívějších vodotisků se používá pro ukrytí jména společnosti, vlastníka nebo majitele.

Podobně jako steganografie je používán k ukrytí informací. Liší se ale od steganografie tím, že u steganografie nesmí být nikdy známa přítomnost skrytého textu tomu, komu není určen. U vodotisku je možné zobrazit informaci viditelně. Steganografie také není odolná nebo má omezenou odolnost. Snaží se skrýt informaci v nějakém médiu tak, aby se nedalo zjistit, že tam nějaké informace jsou. Vodotisk je navržený tak, aby byl odolný. V obrázku může být viditelný nebo neviditelný.

Rozdělení vodotisků

- **Křehký** - má nízkou odolnost vůči změnám[4], stačí malá změna pro zničení vložené informace. Je vytvořen právě kvůli prolomení. Pokud dojde k jeho zničení, bude dokazovat, že se někdo pokusil o útok.
- **Odolný** - většinou je pravým opakem křehkého [4]. Může být viditelný nebo neviditelný v závislosti na účelu. Tyto vodotisky lze velmi těžko poškodit nebo odebrat.

Odolnosti vodotisku:

- **Odolnost vůči ořezávání** – Vodotisk [4]je vložen na tolik míst v obrázku, že odebráním části obrázku se nemůže odstranit vodotisk.
- **Odolnost vůči roztahování a rotaci** – Nejjednodušší je do algoritmu vodotisku [4] vestavět parametry pro roztáhnutí a rotaci obrázku. Tím povolíme rozdíly při opětovném zkoumání vodotisku.

- **Odolnost vůči kompresi** - komprese [4] obvykle pracuje na nějaké frekvenci obrázku, většinou nízké frekvenci. Pak může být vodotisk vložen na vyšší frekvence. Toto udržuje integritu, když se nižší frekvence „odlupují“ v průběhu komprese pro snižování kvality a velikosti jako je například JPEG komprese.

Typy vodotisku

- **Viditelný:** je velmi odolný. Je vytvářený [4], aby byl na první pohled vidět. Jednoduchý příklad viditelné steganografie je identifikační televizní logo nějaké společnosti, viditelné vpravo nebo vlevo nahoře v televizi. Tento vodotisk může být pevný nebo poloprůhledný. Odebrat vodotisk lze oříznutím části obrázku, v které je. Protože ale dojde ke změně obrazu, bude to vždy nápadné. Viditelného vodotisku bývá použito pro reklamní účely a pro vytvoření jakéhosi pouta s majitelem nebo vlastníkem. Ukázka vodotisku, která je vidět v televizích je na *obrázku 7.*

Obrázek 8: Příklad vodotisku [1]

- **Neviditelný:** Jeho účelem je identifikovat vlastníka nebo majitele nebo ověřit integritu, respektive platnost obrázku nebo části informace. Neviditelný vodotisk je nepatrný, ale lze ho vyjmout aritmetickými metodami. Obyčejně je při vyjmutí vodotisku požadován klíč. Tento klíč se nazývá vodotiskový klíč. Nesmí se zaměňovat s šifrovacím klíčem. Vodotiskový klíč působí pouze na vodotisk, zatímco šifrovací klíč je jen semínko, které působí na celou informaci, která byla zašifrována.

Použití neviditelného vodotisku

Hlavní použití vodotisku[4] v digitálním světě je bezpochyby pro jakýsi podpis vlastníka nebo majitele. Je to situace, kdy by někdo sebral materiál a odebral informace o vlastníkovi. Tento důkaz vlastnictví může mít dvě formy. Může identifikovat vlastníka materiálu nebo může identifikovat toho, komu byl materiál zaslán. Spolu s viditelností nebo neviditelností vytváří čtyři možnosti vodotisku.

Neviditelného vodotisku bývá použito také rafinovaněji. Například může být použit jako jakýsi štítek, který dovoluje najít Váš obrázek na internetu, abyste se přesvědčili, zda nebyl zneužit.
Do vodotisku se také mohou ukládat informace například o jméně, autorovi, datu, a kontaktu. Tyto informace mohou být obyčejně užitečné pro všechny.

Tento typ vodotisku je dobrý pro revizi. Například písnička s neslyšitelným vodotiskem může být použita pro zkoumání, kolikrát byla přehrávána v rádiu, a porovnat tento počet s podepsanou smlouvou. Stejně může být tento vodotisk použitý i pro reklamy v televizi, kolikrát byly přehrávány. Když došlo k rozmachu DVD, filmové společnosti projevily velký zájem o označení svých filmů vodotiskem, aby předešly neautorizovanému kopírování. Označení ve vodotisku, společně s hardwarem umožnilo DVD, aby neslo instrukce jako „nikdy nekopírovat“ nebo „kopírovat jedenkrát“,

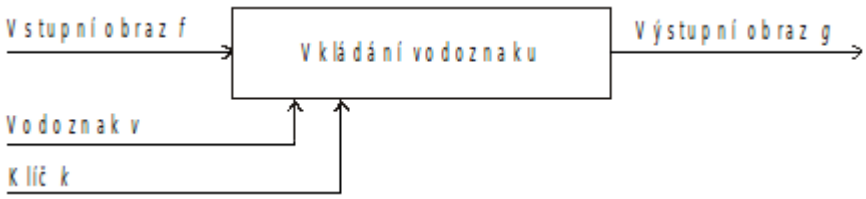
1 převzato z http://www.b-roll.net/today/2009/09/tv-azteca-and-the-watermark/

což umožnilo uživateli vyrobit si jednu záložní kopii, ale ne neomezený počet kopií DVD.

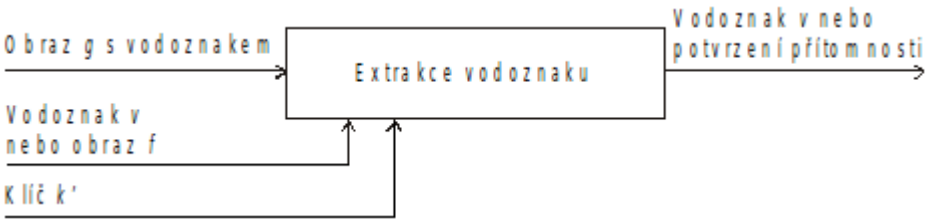
Nejdůležitější vlastnosti vodotisku

- **Nadbytečnost (redundance)** – tato vlastnost [5] určuje, zda jsou informace vodotisku vloženy do obrazu redundantně, respektive opakovaně, tak, aby při vyjmutí vodotisku bylo co nejvíce obtížné. Tato vlastnost určuje také odolnost vodotisku například vůči ořezávání nebo otáčení. Také se rozlišuje, zda se jedná o křehký vodotisk, kde ve většině případů chceme, aby bylo poznat, že bylo manipulováno z obrázkem. Nebo o odolný vodotisk, u kterého chceme, aby nedošlo z poškození a proto vkládáme informace vodotisku opakovaně na různá místa.
- **Kryptografická ochrana** – Vlastnost [5], která určuje, zda může uživatel vyjmout (extrahovat) vodotisk, zda ho může alespoň detekovat či modifikovat. Například, že by mohl podvrhnout svůj, nebo cizí vodotisk a pak by nebylo možné zjistit původního majitele média.
- **Neviditelnost** – tato vlastnost [5] je potřebná, pokud uživatel média nemá vůbec vědět o tom, že médium nějaký vodotisk obsahuje. Ovšem ukrytý vodotisk nesmí výrazně zhoršit kvalitu obrazu, protože pak by jej bylo možné spatřit. Tato vlastnost se týká nejen případů, kdy je vodotisk neviditelný, ale i případů, kdy je vodotisk viditelný.

Obrázek 9 znázorňuje vkládání a *obrázek 10* znázorňuje vyjímání vodotisku [1].



Obrázek 9:Vodotisk používá klíč k a informaci vodoznaku v pro vložení vodoznaku do vstupního obrazu f,a tím vytvoří výstupní obraz g.



Obrázek 10: Vodotisk používá klíč k a informaci vodoznaku v k vyjmutí, nebo alespoň potvrzení přítomnosti vodoznaku v obrazu s vodoznakem g.

Použitý klíč určuje pozici, kde se budou informace vodoznaku nacházet a co s nimi bude možné dělat.

Dělení vodotisků, podle způsobu jejich extrakci

Při extrakci vodotisku záleží na tom, která metoda byla použita pro vložení vodotisku [1].

- **Veřejné vodotisky** – tyto vodotisky jsou nejlepší, bohužel jsou implementačně velmi složité. Jedná se metodu extrahující vodotisk z obrazu bez znalosti originální obrázku před vložením

vodotisku. Výsledkem je samotný extrahovaný vodotisk. Tyto vodotisky mohou mít kryptografický klíč jako vstup. Zde je definičním oborem extrakční funkce $G \times K \to V$.

- **Polosoukromé vodotisky** – jsou podobné soukromým vodotiskům. Sice je potřeba extrakční funkce pro detekci vodotisků, ale k samotné extrakci není třeba používat originální obraz. Extrakční funkce tedy může pracovat s vodotiskem ukrytým v obraze, samotným vodotiskem nebo pouhým klíčem. Výsledek detekce řekne, zda obraz obsahuje daný vodotisk nebo ne. V tomto případě mají extrakční funkce definiční obor $G \times K \times V \to \{0,1\}$.
- **Soukromé vodotisky** – u tohoto případu je nutné pro extrakci vodotisku použít originální obraz. Tento originální obraz slouží pro určení změn. Originální obrázek a obrázek s vodotiskem se porovnají, a výsledkem jsou rozdíly mezi oběma obrázky. Oblasti s rozdílovými hodnotami mohou označovat místa, kde se ukrývá vodotisk nebo kde došlo k poškození obrázku v důsledku přenosu nebo jeho editace. Pokud tato informace stačí pro extrakci vodotisku, jedná se o jednodušší metodu extrakce vodotisku, která má definiční obor extrakční funkce $F \times F \times K \to V$. Pokud tato informace nestačí, jedná se o složitější metodu, kde je potřeba samotný vodotisk. Touto metodou lze jen zjistit, zda obraz vodotisk obsahuje, či nikoliv. Tyto detekční funkce mají definiční obor extrakční funkce $F \times F \times K \times V \to \{0,1\}$.

Metody vkládání vodotisku

- **Vkládání metodou LSB modulace** - tato metoda [4] je stejná jako u předešlé steganografie. Je to metoda založená na modifikaci nejméně významného bitu každé hodnoty pixelu. Při této změně by neměla nastat v obraze změna, která by byla viditelná pouhým okem. V nejnižším bitu každého pixelů se může ukrývat bit, který je částí ukrývaného vodotisku. Detekci těchto pixelů můžeme ztížit tím, že vygenerujeme vodotisk pseudonáhodně, ve formě šumové funkce. Bohužel tato metoda patří mezi málo odolné vodotisky, protože neviditelný vodotisk lze odstranit libovolnou operací s obrazem.
- **Vkládání vodotisku do Y kanálu** -o této metodě [4] jsem hovořil již ve steganografii. Lze ji aplikovat pouze na barevné obrázky. Lze říci, že je to vylepšená metoda LSB. Barevný model RGB originálního obrázku se převede do jiného barevného modelu, v tomto případě do YUV. Poté se do nejnižšího bitu složky Y, vloží bit informace. Pak je barevný model převeden zase nazpět. Lze použít i jinou metodu ukládání dat. Důležité je samo převedení barevného modelu z jednoho, do druhého. Lze také použít jinou složku modelu – U nebo V. Protože informace není přímo viditelná jako u metody LSB, je obtížnější ji najít.
- **Vkládání vodotisku do množin pixelů** - tato metoda [4] patří již mezi pokročilejší metody pro vkládání vodotisku do obrazu. Vybereme-li náhodně dvě stejně velké množiny pixelů z obrazu, bude pravděpodobné, že si budou součty hodnot pixelů z těchto dvou náhodně vybraných množin blízké. Pokud vložíme vodotisk a ovlivníme tím hodnoty těchto pixelů tak, aby součet hodnot pixelů z první množiny byl větší než součet hodnot pixelů druhé množiny, bude toto znamenat log. 0. Pokud součet hodnot z první množiny z první množiny byl menší než součet hodnot druhé množiny, bude to znamenat log. 1. Pak lze pomocí dvou množin pixelů vyjádřit bit skryté informace. Pokud pak použijeme více skupin pixelů, lze ukrýt takto celou zprávu, celý podpis. Abychom ovlivnili součet množin pixelů, odečteme nebo přičteme ke každé hodnotě 1, de požadovaného výsledku. Tyto množiny generujeme náhodným výběrem pixelů z celého obrázku. Není nutnou podmínkou, aby množiny pixelů byly disjunktní. Ale pokud jsou disjunktní, účinnost tohoto algoritmu zvýší.
- Vhodnou metodou pro vodotisk je i **patchwork** [4], zmíněný v části věnované steganografii. Tento algoritmus vkládá do originálního obrázku speciální statistickou informaci, respektive

malý vodotisk, který říká, jak velký vodotisk je ukryt v obrázku. Je to indikátor, kterým se určí, zda je v obrázku vodotisk, či ne. Jistým vylepšením patchworku je například rozmístění více takových informací po celém obrázku, aby nedošlo k jejímu poškození. Patchwork je velmi odolný vůči ořezání a korekci odstínů barev. Není odolný proti roztažení a otočení obrázku. Protože obsahuje velmi malé množství dat, je odolnější než většina ostatních vodotisků. Proto je velmi vhodné ho použít pro podpis digitálních médií, jako jsou obrázky nebo multimediální soubory.

- **Ortogonální projekce** - tato metod [4] je vhodná pouze pro digitální vodotisk, S obrázkem je manipulováno takovým způsobem, že tajná data jsou ukryta do nejdůležitějších částí obrázku, a proto je menší možnost jejich poškození nebo zničení v důsledku běžné manipulace, nebo komprese. Protože tato metoda používá ke své práci nejdůležitější část obrázku, jádro obrázku, mnohem lépe odolává vodotiskovým útokům nebo ztrátové kompresi .

Velmi výstižný popis struktur jednotlivých formátů obrázků, je uveden ve článcích [7] [8] [9] [10].

3. Zamyšlení nad současným stavem

V této kapitole se zamýšlím nad moderními metodami steganografie, které byly vysvětleny v minulé kapitole a také se pokouším v podkapitole 3.2 navrhnout požadavky pro budoucí aplikaci.

V našem elektronickém světě jsou tři způsoby[4] ukrytí ukrýt digitální zprávy (kapitola 2), ale pro obrázky jsou použitelné jen 2 z nich. Metoda *vstřikování (injection)* nebo *substituce (substitution).* Myslím si, že metoda injektivní není moc dobrá z důvodu, že výsledný obrázek je větší než původní, a tím se stává nevýhodným, protože u takového obrázku by si mohl každý všimnout větších rozměrů obrázku, které jsou v obrázku nadbytečné.

Myslím, že pro ty, co chtějí ukládat více informací do obrázku, bude nejlepší metoda LSB. Tato metoda totiž poskytuje optimální poměr velikosti uložené informace / bezpečností metody, jak bylo již zmíněno v *kapitole 2*. Tato metoda sice z důvodu ukrývání dat přímo do pixelů neoplývá přílišnou bezpečností, ale použije-li se vhodná metoda skrytí dat, pak nelze informace najít.

Bohužel se tato metoda nedá použít u nejpoužívanějších obrázků typu JFIF, respektive s příponami jpg a jpeg, protože při opětovném uložení takového souboru, dojde k přepočítání pixelů DCT (diskrétní kosinovou transformací) a tím se vyhladí všechny změny v pixelech. Musí se zde ukládat do parametrů Fourierovy nebo kosinové transformace, a do nich se vejde mnohem méně dat, při ukládání dat do samotných pixelů. Naproti tomu, protože neukládáme informace přímo do pixelů, je tu mnohem větší bezpečnost. Další výhodou je, že pokud bychom měli film formátu MPEG, u kterého lze říci, že používá snímky JFIF navrstvené na sobě, mohli bychom uložit do každého takového snímku informace. Pokud pokud by ten film používal snímkovou rychlost PAL, neboli 25 snímků/s a trval 30 minut, dostali bychom celkem 60*30*25 snímků, což je 45000 snímků. Pokud do každého snímku uložíme 10 bytů informace, dostaneme celkem 450000 bytů informace, což je 450000 znaků kódovaných ASCII tabulkou, pokud nepočítáme informaci o vlastnostech steganografie, která je v takovýchto kódech většinou uložená [4].

Nejefektivněji tuto metodu pravděpodobně využije obrázek gif, který může mít v rámci animace více vrstev obrázků na sobě. Škoda jen, že do obrázku typu GIF lze uložit do každého pixelu jednu z pouze 256 barev. To je velmi málo. Při změně více pixelů za sebou, by mohl teoreticky vyniknout jiný odstín barvy, a napohled by už obrázek nevypadal nevinně. Poté je problémem, kolik bitů je vlastně možné do obrázku uložit. Maximální práh každé metody je jiný,a obecně koresponduje s velikostí obrázku a s hodnotami pixelů.

Další metodou je ukládání do parametrů, kterou využívají obrázky typu JFIF. V těchto obrázcích lze uložit velmi málo informace, ale pokud bychom měli k dispozici film formátu mpg, který používá vlastně snímky jpg navrstvené na sobě. Pokud bychom měli film o délce 30 minut ve formátu PAL, měli bychom 60*30*25 snímků, což je 45000 snímků. Pokud budeme ukládat do každého snímku asi jen 50 bitů, je stále spoustu místa i pro velké zprávy.

3.1.Programovací jazyk

Pro vytvoření aplikace by bylo vhodné si zvolit programovací jazyk a třeba i grafický toolkit pro tvorbu grafické aplikace. Myslím, že nejlepší cestou bude si vybrat jazyk C++, protože se dost často používá. Je to výborný stabilní jazyk, který obsahuje svobodné knihovny, stále se inovuje novými standardy a není proprietární, tudíž může fungovat velmi rychle na každé distribuci. Poskytuje objektově orientované programování, a proto je vhodný i pro týmové návrhy. Také je to jazyk který se mi zalíbil. Ostatní jazyky s kompilátorem jako například JAVA nebo C# by také mohly být dobré, ale já jim nevěřím, protože většinou nejsou stabilní a zakládají se na proprietárních funkcích. Myslím, že

jedinou jejich výhodou je snad jen to, že používají jednu kompilaci, pro všechny druhy operačních systémů a proto není nutná rekompilace při přechodu ze systému na systém. S jazyky, používající interpretr nepočítám, protože pro metody steganografie a pro samotnou práci s obrázkem by byly moc pomalé.

Pro grafické prostředí bude dobrý QT, protože je velmi známý a existuje pro něj výborné vývojové grafické prostředí pro tvorbu projektů - QT Creator. Navíc má spoustu knihoven, které usnadňují práci s obrázky, a to je v tomto případě velmi potřebné.

3.2 Souhrn požadavků na metodu a aplikaci

Mým cílem je vytvořit aplikaci, která by splňovala následující požadavky.

Souhrn požadavků na aplikaci:

- **Formát –** aplikace by měla podporovat PNG, BMP formát a pokud to půjde, i JFIF formát (JPEG/JPG).
- **Grafické rozhraní** - počítám s tím, že kromě steganografických obrázků by měla mít moje aplikace jako výstup nějaké porovnání steganografického obrázku s originálem. Grafické toolkity OO jazyky poskytují také většinou knihovny pro práci s obrázky, které by mohly práci výrazně ulehčit.
- **Jazyk –** změna jazyka není v požadavcích, ale mohlo by to být užitečné pro uživatele, kteří neznají češtinu.
- **Obrázek** - mělo by být možné zvolit obrázek, z kterého budou čteny data a tyto data se budou s vloženými tajnými informacemi zapisovat do samostatného souboru.
- **Šifrování** - mělo by být možné zvolit možnost šifrování nebo nešifrování informací.

Souhrn požadavků na samotnou metodu.

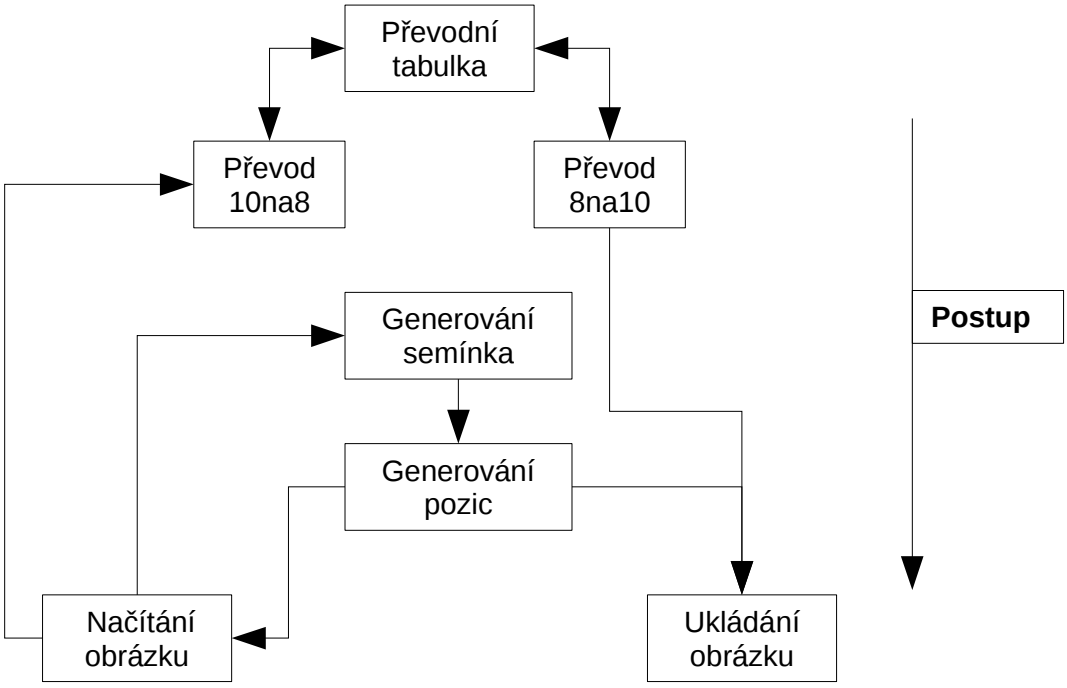
- **Nezávislost metody na formátu obrázku** – pro širší použití nové metody
- **Pozice pixelů** – pozice pixelů, které budou uloženy by měly být vygenerovány pomocí semínka, které bude vytvořeno z informací ze souboru.
- **Vyrovnání bitů** – vyrovnání počtu 0 a 1 u zapisovaných bitů a tím zvýšit bezpečnost ukrytí informace.
- **Ukládání znaků po bytech** – možnost ukládat celou ASCII tabulku, později třeba i binární soubory
- Výsledný obrázek by měl vypadat stejně jako originální, co se týče pixelů, ale také vlastností souborů

Nakonec pro srovnání této nové (navržené) metody implementuji i některou z běžně používaných steganografických metod, uvedených v kapitole 2. Nejsem si totiž jist, zda by některý program pro detekci steganografie dokázal detekovat tuto metodu a proto provedu alespoň toto srovnání.

4. Realizace

V této kapitole probírám samotný návrh mého řešení. Je zde pojednáváno o těchto částech:

1) Obecný návrh
2) Výpočet semínka
3) Princip jednoduché šifry
4) Převodní tabulka
5) Zbytek částí
6) Implementace a příklad funkcí
7) Úprava metody sousedů
8) Obrázky aplikace



Šipky představují komunikaci mezi prvky

Diagram 1: Vývojový diagram projektu

Diagram 1 ukazuje postup při tvorbě aplikace (šipka vpravo). Ostatní šipky ukazují propojení mezi jednotlivými částmi aplikace. Bližší vysvětlení poskytuje *Pseudokód 1*.

1) Nová metoda vychází z dříve zmíněné metody LSB. Částečně užívá metodu vodotisku (kapitola 2). Tato metoda funguje na principu generátoru pseudonáhodných čísel. Tento generátor generuje pozice pixelů v konkrétním obrázku. Do těchto pixelů bude ukryta skrytá informace.

2) Tzv. semínko, použité pro generování pseudonáhodných čísel se generuje z vlastností obrázku. V mém případě jsem zvolil 3 bity šířky, 3 bity výšky, bitové informace o tom, jestli je obrázek ve stupních šedi, zda má tzv. alfa kanál, a nakonec 2 bity bitové hloubky obrázku (Text1). Tyto bity jsou promíchány mezi sebou, a výsledné seskupení bitů tvoří číslo, kterého je použito jako semínka.

Bitové znázornění výsledného semínka
HWHWGDDAWHWHWH

H – výška, popořadě 10,9,7,6,2,0 bit z výšky
W – šířka, popořadě 11,8,5,4,3,1 bit ze šířky

D – bitová hloubka
Pokud je 8, nastaví se 7 bit
Pokud je 16, nastaví se 8 bit
Pokud je 24, nastaví se 7 i 8 bit
V ostatních případech zůstává na nule.

G – je v stupních šedi?
A – má alphachannel?

Text 1: Systém počítání semínka

3) Informaci, která má být skryta, je možné v nové metodě šifrovat pomocí jednoduché šifry po jednotlivých bytech. Tato jednoduchá šifra je složena z negace, zpětné ekvivalence a nonekvivalence. Při tvorbě tohoto jednoduchého systému šifrování jsem dbal na to, aby žádný z 256 znaků ASCII abecedy neměl po překódování stejnou hodnotu.

Šifrování:

1. Provede se ekvivalence bitů bi a bi+1 od LSB bitu po MSB, pro všech 8 bitů. Nejmenší bit se zapíše na nejmenší pozici výsledku, aby měl opět 8 bitů
   Příklad: 01011010
   Výsledek: 00010000

2. Provede se nonekvivalence mezi MSB a LSB bitem. Tento výsledek se zapisuje zleva doprava. Poté se provádí totéž s MSB-1 bitem a předchozím výsledkem. Provádí se, dokud se nedostaneme za LSB bit
   Příklad: 01011010
   Výsledek: 01101100

3. Provede se negace všech 8 bitů
   Příklad: 01011010
   Výsledek: 10100101

Dešifrování probíhá stejnými třemi metodami, ale v obráceném pořadí

Pseudokód 2: Princip šifrování a dešifrování

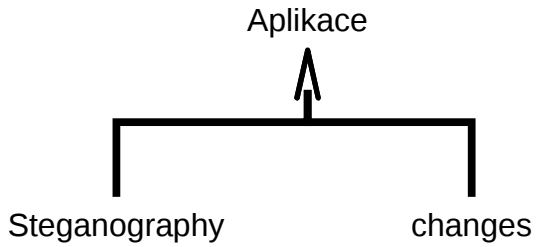
4) Dále je použit systém kódování 8 na 10, sloužící k odstranění nerovnoměrného počtu jedniček a nul. Pro konverzi 8 na 10 je použita vlastní funkce, aby bylo ztíženo dekódování neoprávněnou osobou. Tento způsob kódování snižuje množství informace, které je možné uložit do obrázku, ale tuto oběť nahrazuje výrazně vyšším zabezpečením dat.

5) V této nové metodě je možné uložit do obrázku text po bytech. Toto umožní uložit stejnou zprávu pomocí celé ASCII tabulky, ale také možné budoucí rozšíření nové metody pro ukládání binárních souborů, protože kódování v osmicích bitů je nejjednodušší pro ukládání hexadecimální informace.
   Počítám, že bezpečné bude ukládat data pouze na 10% pixelů obrázku = šířka obrázku* výška obrázku/10. Generování většího počtu pozic by vedlo ke zpomalení činnosti programu v důsledku kontroly, zda vygenerovaná pozice již nebyla předtím užita. Počítám také, že pokud se bude ukládat informace, některé pixely se vůbec nezmění. Tím by se mohl snížit výskyt šumu, ovšem v závislosti na konkrétním obrázku a konkrétních ukládaných datech.
   Konkrétní ukládání do jednotlivých pixelů je provedeno za použitím standardních knihoven toolkitu qt pro otvírání a editace obrázků. Použitím standardních knihoven je dosaženo velké rychlosti zpracování obrázků. Vygenerování převodní tabulky probíhá již při spuštění aplikace. Pseudokód algoritmu je na *Pseudokód 1*.
   Nová metoda pracuje s černobílými obrázky stejně jako s barevnými, proto bylo nutné zvolit, která barevná složka ponese informaci. Zvolil jsem ukládání do všech třech barevných složek postupně. Např. První vygenerovaný pixel ponese informaci v červené barevné složce, druhý vygenerovaný pixel ponese informaci v zelené barevné složce a třetí v modré barevné složce. Pro další pixely se tento postup opakuje.

Načítání informace z obrázku a následný zápis informace
1. Sestavení převodní tabulky
2. Načtení obrázku
3. Výpočet semínka z obrázku
4. Generování pozice pixelu ze semínka, byla-li vygenerována předtím, skok na 4.
5. Načtení a uložení LSB bitu z pixelu
6. Pokud nebylo načteno 10 bitů skok na 4.
7. Převod 10 bitů na 8 bitů
8. Neoznačuje-li 8 bitové číslo koncový bod uložení dat a návrat na 4.
9. Nebyla-li nalezena 8 bitová převodní kombinace, konec (obrázek neobsahuje steganografii)
10. Obsahuje-li koncový bod informaci o šifrování, dešifrování převedených čísel
11. Zobrazení výsledné informace a konec
12. Načtení informace od uživatele, pro uložení do obrázku
13. Vygenerování tolika různých pozic, kolik je potřeba pro uložení informace
14. Je-li požadováno šifrování, jednotlivé byty informace se zašifrují
15. Převod jednotlivých bytů informace na 10 bitů
16. Uložení jednotlivých bitů informace na vygenerované pozice
17. Uložení výsledného obrázku

Pseudokód 1: implementace navržené metody

6) Celý můj program je složen ze dvou tříd. Důležitá je hlavně první třída „Steganography“. Druhá třída „changes“, je poděděna ze standardní třídy QWidget a slouží k vykreslování změn mezi původním (originálním) obrázkem a novým (steganografickým) obrázkem. Hlavní třída „Steganography“ slouží pro zapisování a načítání obrázků pomocí steganografických metod.



Třídy aplikace

Samotné vykreslování probíhá v konstruktoru „changes()“, a to tak, že nejprve se vytvoří obrázek změn, který má stejné rozměry, jako mají originální i steganografický obrázek. Poté se vyhodnotí, které pixely v originálním po(x,y) a steganografickém obrázku ps(x,y) mají vůči sobě různé hodnoty RGB a tyto pixely jsou podle počtu změněných bitů vykresleny jinou barvou. Více v kapitole 5.

Funkce „void createtbl()“, která vytváří tabulku s převodními hodnotami, pro převod z 8 na 10 bitů a nazpět. Počet nul = počet jedniček má 252 čísel, 8 čísel je s 6 nulami a 4 jedničkami, nebo naopak. Také tu jsou 4 čísla obsahující informaci o steganografii. Jedno z těchto čísel se řadí na konec steganografického textu. Tyto čísla označují, zda se text šifruje / nešifruje kryptografií.

Příklad funkce aplikace

Rád bych zmínil odlišnost ukládání ve formátech PNG a BMP a také modifikaci této metody pro JPEG metodu. Odlišnost ukládání je v tom, že u obrázků typu PNG přepisuji i některé chunky, tvořící hlavičku souborů, chunky z původního (originálního) obrázku. Je tomu tak proto, že výsledný soubor, zpracovaný pomocí standardních knihoven QT má zcela odlišnou hlavičku. A to nesplňuje požadavky, kladené na metodu.

Modifikace nové metody pro JPEG principiálně funguje naprosto stejně, ale generuje se pouze jedno číslo, ne pozice (2 čísla pro X a Y). Toto číslo je použito pro určení pozice v tabulkách parametrů kosinové transformace, kam bude uložen 1 bit informace. Takto se pokračuje dále, dokud stačí bity informace.

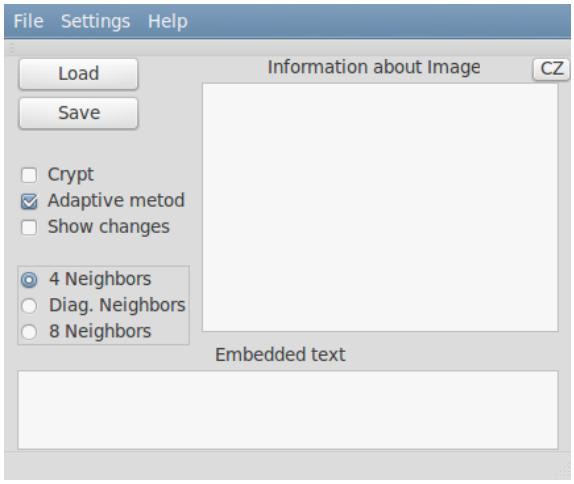
7) Při testování metod 4, 8 a diagonálních sousedů jsem narazil na chybu v podmínkách z kapitoly 2, které jsou zapotřebí pro testování, zda bude pixel použit pro vkládání informace. Opravil jsem je na tento tvar, aby byly funkční. Předešlé podmínky zjevně nepočítaly s tím, že by některý pixel mohl mít hodnotu 0 a okolní pixely průměrnou hodnotu větší než 8.

Protože tato metoda je z principu pro černobílé obrázky, barevné složky se vybírají u každého pixelu stejně jako u navržené metody, totiž postupně R nebo G nebo B u každého dalšího pixelu.

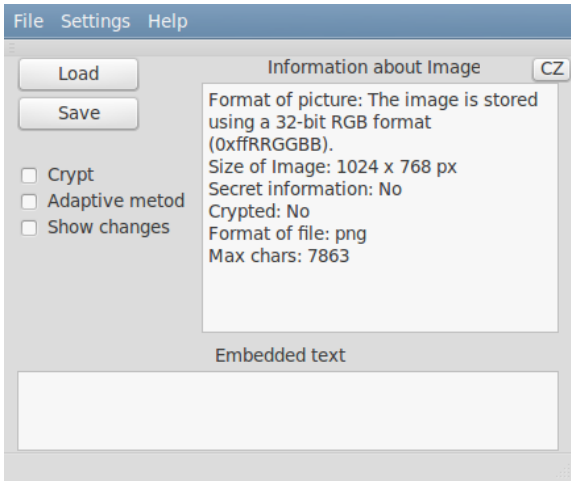
$$(1)\ \frac{p_{nahoře}+p_{dole}+p_{vlevo}+p_{vpravo}}{4}<2^{n+1}-1,\ pak\ pi'=pi,\ resp.\ pixel\ se\ nezmění$$

$$(2)\ \frac{p_{nahoře}+p_{dole}+p_{vlevo}+p_{vpravo}}{4}+2^{n+1}>255,\ pak\ pi'=pi,\ resp.\ pixel\ se\ nezmění$$

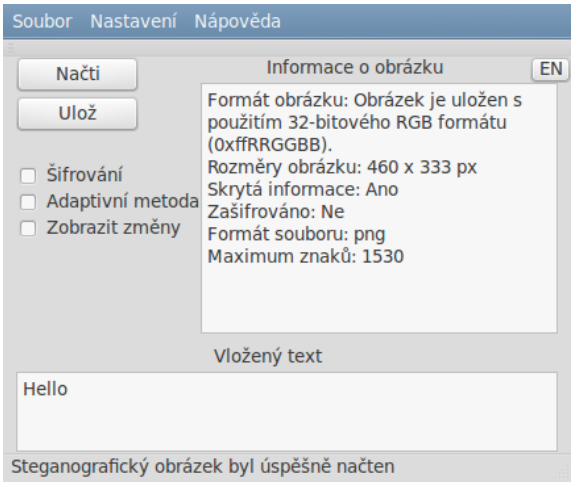
8) Obrázky aplikace



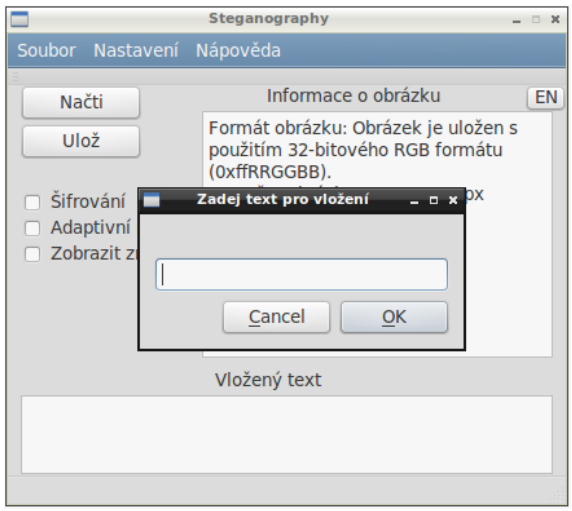
Obrázek 4.1. Anglická lokalizace, adaptivní metoda



Obrázek 4.2. Anglická lokalizace, obrázek bez skryté informace



Obrázek 4.3. Česká lokalizace, obrázek se skrytou informací



Obrázek 4.4. Česká lokalizace, vložení textu pro vložení do obrázku

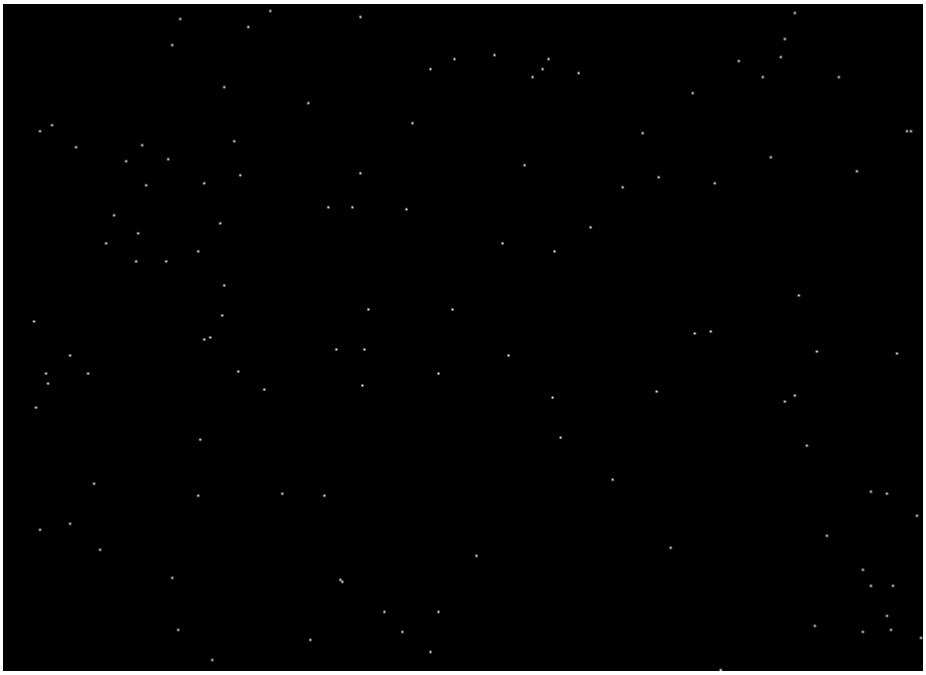
5. Experimenty

V této kapitole jsou ukázány rozdíly mezi novou (navrženou) metodou a metodou sousedů.
Uložený text je: „*Example text for embedding*“

5.1. Navržená metoda

Obrázek 5.1 Originální obrázek

Obrázek 5.2 Steganografický obrázek



Obrázek 5.3 Rozdíly mezi oběma obrázky

Černé pixely – nebyl změněn žádný bit
Bílé pixely – byl změněn 1 bit

Žluté pixely – změny 2 nebo 3 bity
Oranžové pixely – změněno 4 nebo 5 bitů
Červené pixely – změněno více jak 5 bitů

5.2. Metoda sousedů

Obrázek 5.4 Originální obrázek *Obrázek 5.5 Steganografický obrázek*

Obrázek 5.6: Rozdíly mezi originálním a steganografickým obrázkem, metoda 8 sousedů

Bílé pixely – nebyl změněn žádný bit
Černé pixely – byl změněn 1 bit

Žluté pixely – změny 2 nebo 3 bity
Oranžové pixely – změněno 4 nebo 5 bitů
Červené pixely – změněno více jak 5 bitů

5.3. Shrnutí a vysvětlení experimentů

Mezi rozdíly je patrné, že nově navržená metoda používá širší rozložení informace a proto tajná informace nebude nikde příliš viditelná. Když se použije tentýž obrázek na metodu sousedů, tato metoda zaznamená data jen v prvních pixelech v řádku, protože pixely nemají hodnotu okolo 255 nebo okolo 0. Tento obrázek jsem sem nezařadil, protože pixely při okraji nebyly příliš patrné. Pokud ale použiji obrázek, který je černý, tzn. hodnoty pixelů má kolem 0, rozložení informace se změní. Metoda sousedů lépe využije pixely, ale pokud by byl celý obrázek bílý nebo celý obrázek černý, žádnou informaci by neuložila. Také to, že hromadí informace jen v některých oblastech, může vést k odhalení steganografie v obrázku. Navržená metoda takto informace nehromadí. Naopak používá způsob náhodného rozložení, který vytváří šum, který se může objevit v klasickém obrázku při jeho zmenšení a následném zvětšení nebo naopak. Toto potvrzuje i *výpis 1*. Další grafické výstupy metod jsou obsaženy v příloze.

Obrázek kočky 460 x 333 px	*Obrázek catwoman 800 x 600 px*
Max znaků v metodě 4 sousedů: ***7519***	*Max znaků v metodě 4 sousedů:* ***4380***
Max znaků v metodě diag. sousedů: ***7515***	*Max znaků v metodě diag. sousedů:* ***4003***
Max znaků v metodě 8 sousedů: ***3769***	*Max znaků v metodě 8 sousedů:* ***2146***
*Max znaků v představené metodě:****1530***	*Max znaků v představené metodě:****7863***
Max znaků v jpeg modifikaci: ***11***	*Max znaků v jpeg modifikaci:* ***11***

Výpis 1: Výpis informací z předchozích obrázků

Je možné spatřit, že pokud je obrázek černý a velký, má navrch navržená metoda. Pokud je ale obrázek malý a neobsahuje tolik černých nebo bílých pixelů, má navrch metoda sousedů. Obrázky jpeg modifikaci jsou přiloženy v příloze. Je patrné, že její výsledky testování osou obrázků jsou stejné.

Bylo by proto zajímavé tyto metody propojit a tím získat metodu, která by ukládala větší množství bitů na pixel na náhodně vybrané pixely.

Pokoušel jsem se najít také některé programy pro detekci steganografie a testovat jimi steganografické obrázky. Bohužel ani nejznámější program pro detekci, s*tegdetect* nedokázal najít žádnou skrytou informaci. Jeho starší verze, *stegdetect 0.4*, mi ale pomohla najít chybu v (Text2) modifikaci navržené metody pro metodu JPEG.

```
Corrupt JPEG data: 18 extraneous bytes before marker 0xc4
steg-6009.jpg : error: Invalid JPEG file structure: SOS before SOF
```

Text2: Chybový výstup programu stegdetect 0.4

Všechny experimentální obrázky pocházejí z www.google.com

Při experimentování jsem se setkal s chybami. Tyto chyby se mi podařilo opravit a tak ověřit funkčnost celé aplikace

6. Závěr

Cílem této práce bylo navrhnout metodu pro skrývání informací do obrázků a pro tuto metodu vytvořit aplikaci. Tento cíl byl splněn.

V průběhu řešení práce jsem se seznámil s problematikou vkládání informace do obrázku. výsledky jsou shrnuty v kapitole 2

Byl navržen vhodný způsob ukládání informací do obrázku. Tato metoda steganografie je specifikována v kapitole 4.

Pro tuto novou metodu byla navržena aplikace. Obrázky této aplikace a obecné informace o ní jsou uvedeny v kapitole 4. Zdrojové kódy aplikace se nachází na přiloženém CD.

Nově navržená metody je funkční, což bylo demonstrováno na implementované aplikaci a jejich výsledcích. U navržené metody je použito jen 10% celkové kapacity obrázku a do pixelu se ukládá jen 1 bit (navíc jen do jedné barevné složky). Pokud bychom použili plnou kapacitu obrázku a ukládali bychom více bitů na každý pixel, bylo by možné kapacitu zvýšit několikanásobně, možná však na úkor bezpečnosti.

Snažil jsem se vytvořit aplikaci, která by dokázala do obrázku vložit informaci tak, aby obrázek zůstal co nejvíce stejný co do vzhledu, tak i vlastností obrázku. Toto se mi, ale podařilo jen částečně, protože soubory typu PNG používají kódování zlib pro komprimaci dat.

Původně jsem také chtěl využít kapacity obrázků GIF, které mohou obsahovat několik obrázků za sebou. Bohužel QT 4.7 zatím neumožňuje jejich ukládání.

V průběhu tvorby této práce jsem se seznámil se velice zajímavou bezpečnostní technikou pro vkládání informací do obrázků. Také jsem se seznámil podrobně s mnoha formáty obrázků, které jsem poté použil ve své práci.

Možné budoucí rozšíření rozšíření této aplikace může být třemi způsoby. Lze používat soubory s MPEG kódováním, které se skládají z jednotlivých snímků JPEG. Stačí tedy pouze implementovat část, která bude soubory s MPEG kódováním rozkládat na jednotlivé snímky. Možným rozšířením by mohlo být vkládání i binárních souborů namísto textu. Dále by bylo možné propojit obě implementované metody do jedné metody, která by na místa, vygenerovaná s použitím semínka, ukládala data metodou sousedů. Tedy více jak 1 bit do pixelu. Bylo by ale nutné vyloučit sousední pixely u vygenerovaného pixelu z dalších generovaných pozic, jinak by uložená data mohla být zničena.

CITACE A BIBLIOGRAFIE

[1] Dobeš Michal, Zemčík Pavel a kol.: Zpracování obrazu a algoritmy v C#, Praha, CZ, BEN, 2008, s.144, ISBN 978-80-7300-233-6

[2] D.-C. Wu, and W.-H. Tsai, “A steganographic method for images by pixel-value differencing,” Pattern Recognition Letters, Vol. 24, pp. 1613–1626, 2003.

[3] Hossain Moazzam, Sadia Al Haque, Sharmin Farhana.: Variable Rate Steganography in Gray Scale Digital Images Using Neighborhood Pixel Information, *Department of Computer Science and Engineering,* IAJIT, January 2010, Vol.7, No. 1, s. 34-38.

[4] Kipper Gregory.: Investigator´s guide to steganography, Florida, CRC Press Company, 2004.Vyd. 1. 220 s., ISBN 0-8493-2433-5

[5] T Morkel, JHP Eloff and MS Olivier, "An Overview of Image Steganography," in Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005 (Published electronically)

[6] Watkins Jonathan.: Steganography - Messages Hidden in Bits. *Multimedia Systems Coursework, Department of Electronics and Computer Science, University of Southampton* . 15-12-2001, s. 1-10. ISSN SO171BJ.

[7] TIŠNOVSKÝ, Pavel.: Programujeme JPEG: Interní struktura souborů typu JFIF/JPEG, Seriál Grafické formáty, Root.cz, 25.1.2007, Dostupné z WWW: <http://www.root.cz/clanky/programujeme-jpeg-interni-struktura-souboru-typu-jfifjpeg/>

[8] TIŠNOVSKÝ, Pavel.: Programujeme JPEG: Načtení informací ze souborů typu JFIF/JPEG, Seriál Grafické formáty, Root.cz, 1.2.2007, Dostupné z WWW: <http://www.root.cz/clanky/programujeme-jpeg-nacteni-informaci-ze-souboru-typu-jfifjpeg/>

[9] TIŠNOVSKÝ, Pavel.: Grafický formát BMP - používaný a přitom neoblíbený, Seriál Grafické formáty, Root.cz, 19.10.2006, Dostupné z WWW: <http://www.root.cz/clanky/graficky-format-bmp-pouzivany-a-pritom-neoblibeny/>

[10] TIŠNOVSKÝ, Pavel.: PNG - bity, byty, chunky, Root.cz, 25.1.2007, Dostupné z WWW: <http://www.root.cz/clanky/png-bity-byty-chunky/>

Příloha A : Další obrázky

A.1. Metoda sousedů

Obrázek A1: Originální obrázek *Obrázek A2: Steganografický obrázek*

Obrázek A3: Rozdíly mezi originálním a steganografickým obrázkem, metoda 4 sousedů

Obrázek A4: Rozdíly mezi originálním a steganografickým obrázkem, metoda 6 sousedů

Text, ukrytý v obrázku je: „Example text for embedding“

Bílé pixely – nebyl změněn žádný bit
Černé pixely – byl změněn 1 bit
Žluté pixely – změny 2 nebo 3 bity
Oranžové pixely – změněno 4 nebo 5 bitů
Červené pixely – změněno více jak 5 bitů

Legenda pro vykreslené změny v obrázku.

A.2. Navržená metoda, modifikovaná pro JPEG

Text, ukrytý v obrázku je: „Hallo world“.

Obrázek A5: Originální obrázek

Obrázek A6: Steganografický obrázek

Obrázek A4: Rozdíly mezi originálním a steganografickým obrázkem

Při přepočítání obrázku kosinovou transformací dojde vždy k obrovské změně.

Černé pixely – nebyl změněn žádný bit
Bílé pixely – byl změněn 1 bit

Žluté pixely – změny 2 nebo 3 bity
Oranžové pixely – změněno 4 nebo 5 bitů
Červené pixely – změněno více jak 5 bitů

Všechny experimentální obrázky pocházejí z www.google.com

Zadání bakalářské práce/11593/2010/xpodho02

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2010/2011

Zadání bakalářské práce

Řešitel: **Podhorský Jiří**
Obor: Informační technologie
Téma: **Steganografie**
Steganography
Kategorie: Bezpečnost

Pokyny:

1. Seznamte se s problematikou ukrývání informací do obrázku.
2. Navrhněte vhodný způsob uchování informací v obrázku. Navržená aplikace musí umožnit skrytí textu bez hesla (např. pomocí jednoznačného identifikátoru) i zaheslovaně.
3. Vytvořte navrženou aplikaci umožňující ukrytí zadaného textu do vybraného obrázku.
4. Diskutujte možná budoucí rozšíření aplikace.
5. Prezentujte projekt plakátkem.

Literatura:

- Dle pokynů a doporučení vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a částečně 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, doc. Dr. Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2010
Datum odevzdání: 18. května 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu